

Grant or Contract # N00014-14-1-0010

Adaptive Meshing of Ship Air-Wake Flowfields

Progress Report for Period: October 1, 2013 – September 30, 2014

PI: Steve Karman

423-425-5492

Steve-Karman@tennessee.edu

University of Tennessee at Chattanooga

Date Prepared: October 21, 2014

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 10/21/2014		2. REPORT TYPE STATUS Report		3. DATES COVERED (From - To) 10/01/2013 - 09/30/2014	
4. TITLE AND SUBTITLE Adaptive Meshing of Ship Air-Wake Flowfields				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER N00014-14-1-0010	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Dr. Steve Karman				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Tennessee at Chattanooga				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research 875 North Randolph Street Arlington, VA 22203-1995				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Public Release					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>Numerical simulations of aircraft landing on a carrier are difficult due to complex geometry and complex flow physics. The flowfield is very unsteady and chaotic and adequate mesh resolution is crucial to a successful simulation. The goal of the proposed research is to deliver enhanced mesh adaptation capabilities that account for the chaotic unsteady nature of the flowfield about an aircraft in the landing approach path. Similar work was published by Shipman, et. al. [Shipman, Arunajatesan, Cavallo, Polsky]</p> <p>The objectives of the research are to explore three distinct mesh adaptation methods to handle the dynamic aspect of this case. The three methods include a hierarchical-Cartesian hexahedral method, an all-tetrahedral mesh method and a physics-based point placement/meshless method. The hierarchical method will subdivide cube-shaped elements to resolve geometry and gradients of user-selected adaptation functions, such as pressure or Mach number. The tetrahedral method is a traditional unstructured mesh method that incorporates adaptation through node movement to resolve gradients of the adaptation function. The third method is a meshless method that uses a physics-based force model to move nodes around to resolve the geometry and flowfield.</p> <p>The initial phase of the research conducted the first year developed steady-state analysis procedures for each method, with appropriate mesh adaptation capabilities. A description of the steady-state version of the three computer codes (TetFlow, OctFlow and PointFlow) is described in this report.</p> <p>The outcome of the research will provide insight into efficient and robust approaches for adaptive meshing for dynamic simulation of aircraft landings in the presence of unsteady carrier flowfield. Research is conducted assuming inviscid flow, but approaches will be applicable to viscous simulations with modifications.</p>					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 27	19a. NAME OF RESPONSIBLE PERSON Dr. Steve Karman
a. REPORT Unclassified	b. ABSTRACT	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code) 423-425-5492

Section I: Project Summary

1. Overview of Project

Numerical simulations of aircraft landing on a carrier are difficult due to complex geometry and complex flow physics. The flowfield is very unsteady and chaotic and adequate mesh resolution is crucial to a successful simulation. The goal of the proposed research is to deliver enhanced mesh adaptation capabilities that account for the chaotic unsteady nature of the flowfield about an aircraft in the landing approach path. Similar work was published by Shipman, et. al. [Shipman, Arunajatesan, Cavallo, Polsky]

The objectives of the research are to explore three distinct mesh adaptation methods to handle the dynamic aspect of this case. The three methods include a hierarchical-Cartesian hexahedral method, an all-tetrahedral mesh method and a physics-based point placement/meshless method. The hierarchical method will subdivide cube-shaped elements to resolve geometry and gradients of user-selected adaptation functions, such as pressure or Mach number. The tetrahedral method is a traditional unstructured mesh method that incorporates adaptation through node movement to resolve gradients of the adaptation function. The third method is a meshless method that uses a physics-based force model to move nodes around to resolve the geometry and flowfield.

The initial phase of the research conducted the first year developed steady-state analysis procedures for each method, with appropriate mesh adaptation capabilities. A description of the steady-state version of the three computer codes (TetFlow, OctFlow and PointFlow) is described in this report.

The outcome of the research will provide insight into efficient and robust approaches for adaptive meshing for dynamic simulation of aircraft landings in the presence of unsteady carrier flowfield. Research is conducted assuming inviscid flow, but approaches will be applicable to viscous simulations with modifications.

2. Activities this period

Three Computational Fluid Dynamics (CFD) codes were developed to serve as test bed for research into adaptive mesh generation techniques. The three codes all solve the unsteady Euler equations, but use different discretization strategies. The target application is an aircraft in a landing approach to a carrier. In general terms, the first year of this project focused on generating the baseline steady-state capability of the three codes running in serial. This will include adaptive mesh generation capabilities for all codes. The second year will address the unsteady moving body/mesh capability. And finally, the third year will explore efficient parallel implementations of the codes. The steady-state capabilities of the three codes (OctFlow, TetFlow and PointFlow) are described below. This is followed by common validation cases analyzed by all three codes.

OctFlow

The Cartesian code OctFlow has capabilities similar to the SPLITFLOW code developed by Lockheed Martin. OctFlow utilizes an Octree data structure and performs cut-cell operations at geometry boundaries. A second-order spatial finite-volume scheme has been incorporated with explicit first order backward time integration. The Cartesian mesh is generated automatically based on the input geometry, which is supplied as a triangulated surface mesh. The cells intersected by the geometry are handled using the “cut-cell” approach, which is basically creating arbitrary polyhedral elements with appropriate surface boundary conditions. Any cells completely outside the computational domain are tagged external and not solved in the flow solution. Examples of this mesh generation process are shown in Figure 1 through Figure 4. Cell gradation is controlled in the mesh generation process. For example, at least two cells of one resolution are enforced before transitioning to a different cell size.

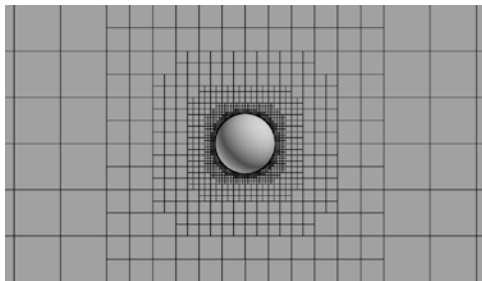


Figure 1. OctFlow mesh for sphere.

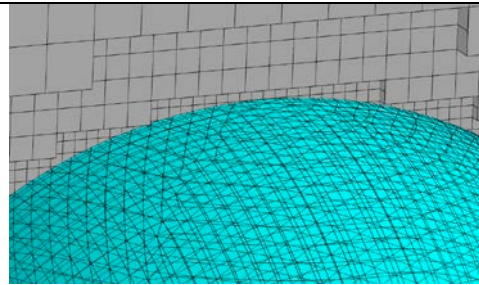


Figure 2. View of surface polygons resulting from cutting.

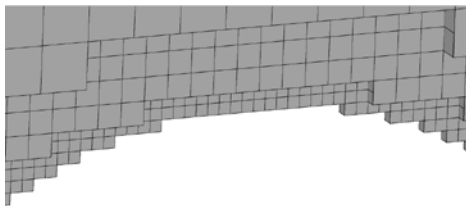


Figure 3. Cartesian cell remain whole.

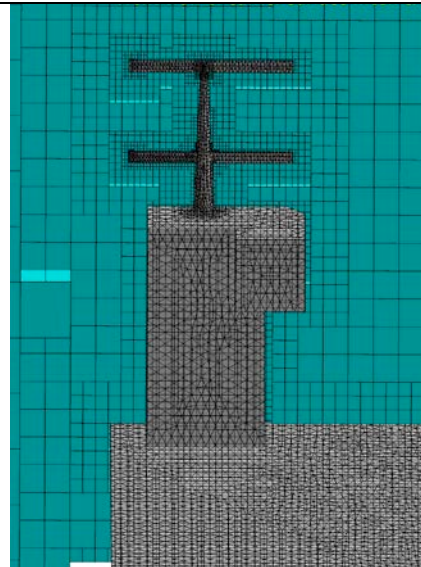


Figure 4. OctFlow mesh near tower of notional aircraft carrier.

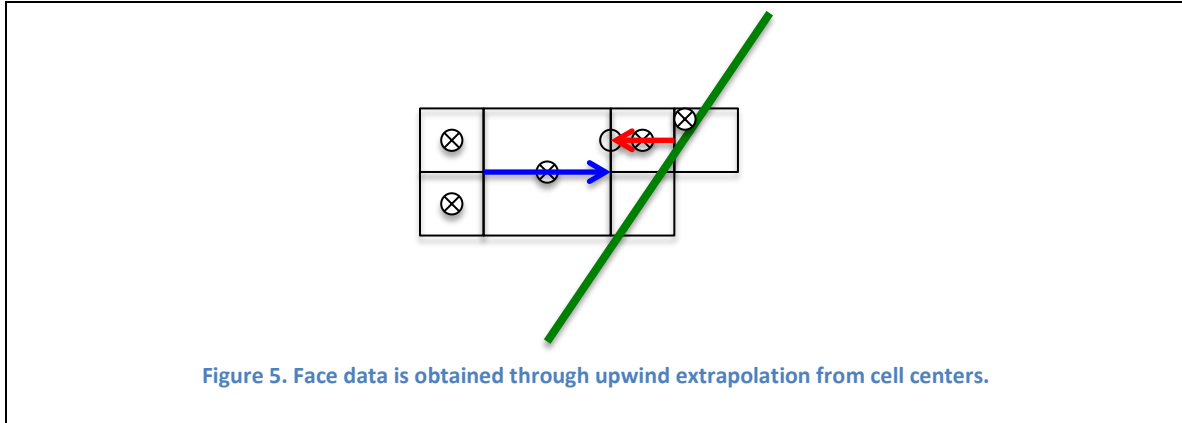
OctFlow solves the integral form of the Euler equations, shown below. This is currently discretized in time using an explicit, first-order backward differencing scheme.

$$V \frac{\partial Q}{\partial t} + \oint_{\partial V} \vec{F} \cdot \hat{n} dS = 0$$

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}$$

$$\vec{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E + p)u \end{bmatrix} \hat{i} + \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (E + p)v \end{bmatrix} \hat{j} + \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (E + p)w \end{bmatrix} \hat{k}$$

Volume averaged flow variables are stored at cell centers, which would be the centroids of the elements. Away from boundaries this would simply be the middle of the cube. Spatial discretization is accomplished using a second-order upwind extrapolation scheme, shown in Figure 5. This is a two-dimensional depiction of extrapolations from cell centers to the element faces. The green line represents a boundary. Primitive variable data is extrapolated in the 3 cardinal directions (X, Y and Z) from both sides of the face. Extrapolations of those “left” and “right” states are limited using a minmod type limiter. The extrapolated, limited values are provided to a Roe Flux Difference Split scheme for the face.



Adaptation is achieved through refinement and de-refinement of the hierarchical Cartesian mesh. The user may select from a current list of four adaptation functions; density, velocity magnitude, pressure or Mach number. Scalar spacing values are computed for each leaf of the mesh (a leaf is the finest level element done each branch of the Octree). That scalar spacing value is combined with an extent box based on the size of the leaf element. This size packet is passed down the

Octree, forcing refinement. The scalar spacing value is computed using the gradient of the adaptation function, shown in the equations below. An element adaptation function, AF , is computed using the magnitude of the gradient of the adaptation function for the element multiplied by a length scale, l , raised to a power p . The length scale is simply the cube root of the element volume. The exponent power is user defined and should be a value greater than one. The mean and standard deviation of AF is computed. Then a de-refinement threshold, AF_d , and a refinement threshold, AF_r , are computed. An element is marked for deletion if its adaptation function and all of its siblings' adaptation function is below AF_d . If an element has an AF greater than the refinement threshold then a spacing value is computed by solving the first equation below for the scalar spacing value (l is replaced with s).

$$AF_e = |\nabla f_e| l_e^p$$

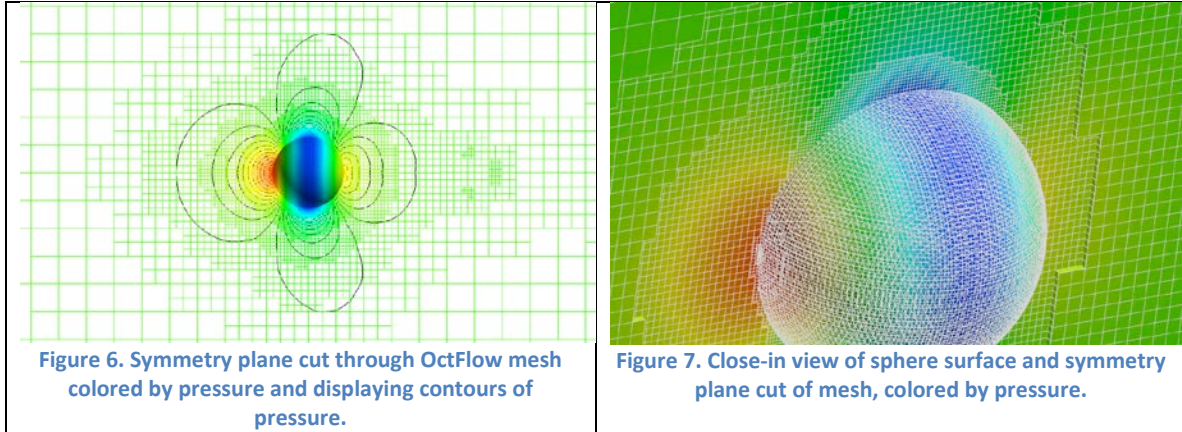
$$l_e = \sqrt[3]{V_e}$$

$$AF_d = AF_{mean} - C_c AF_{STD}$$

$$AF_r = AF_{mean} + C_r AF_{STD}$$

$$s = \left[\frac{AF_r}{|\nabla f_e|} \right]^{\frac{1}{p}}$$

An example of adaptation is shown in Figure 6 and Figure 7. The symmetry plane volume mesh is shown in both figures. The adaptation function was pressure. Finer resolution elements are clearly visible where the gradients of pressure are highest.



The baseline serial capability for OctFlow has been completed. Validation cases are underway. The plans for the next phase include implementing a 2nd order time integration scheme, either BDF2 or some form of hierarchical multi-grid scheme (Octree is a natural platform for multi-grid). In addition the moving body capability will be implemented. This involves grid speed terms for the boundary conditions. The volume mesh does not move. The boundaries move through the

volume mesh, so grid speed terms are not required for the Cartesian mesh. However, certain volume elements will pass in and out of existence, with respect to the solution domain. This must be handled in a time-accurate, conservative and consistent manner.

TetFlow

The code TetFlow has been developed as the test bed for more traditional unstructured solvers. It is a node-centered, finite-volume scheme for meshes comprised of tetrahedral elements only. It solves the Euler equations using second order spacial differencing and up to second order temporal differencing. The discrete set of equations for the implicit BDF scheme is shown below. [Biedron, Thomas] Pseudo-time stepping, $\Delta\tau$, sub-iterations are used to advance the solution in physical time, Δt .

$$\left[\left(\frac{V^{n+1}}{\Delta\tau} + \frac{V^{n+1}\phi_{n+1}}{\Delta t} \right) I - \frac{\partial R^{n+1,m}}{\partial Q} \right] \Delta Q^{n+1,m} = R^{n+1,m} - Q^n R_{GCL}^{n+1} - V^{n+1} \frac{\phi_{n+1}}{\Delta t} (Q^{n+1,m} - Q^n) - V^{n-1} \frac{\phi_{n-1}}{\Delta t} (Q^{n-1} - Q^n)$$

$$R = - \oint_{\partial V} \vec{F} \cdot \hat{n} dS$$

$$R_{GCL}^{n+1} = \frac{1}{\Delta t} [\phi_{n+1} V^{n+1} + \phi_n V^n + \phi_{n-1} V^{n-1}]$$

The Geometric Conservation Law portion (shown in red) has not been implemented yet. Temporal accuracy of the BDF scheme is controlled through the Φ coefficients, shown in the Table 1. [Biedron, Vatsa, Atkins]

Table 1. Coefficients for BDF schemes

Order	Φ_{n+1}	Φ_n	Φ_{n-1}
1st	1	-1	0
2nd	3/2	-2	1/2

The control volume surrounding each node is the median dual comprised of mid-edge, mid-face and cell centroids. Upwind extrapolations of primitive variables to the mid-edges are performed to achieve second order accuracy. Minmod type limiters are used when strong discontinuities are present. The extrapolation gradient is based on weighted averaging or cell gradient from the upwind direction, shown in Figure 8 and Figure 9. Roe's FDS scheme is then used to compute the fluxes at the control volume boundaries.

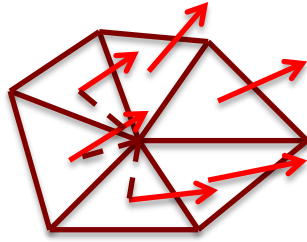


Figure 8. Cell based gradient vector from upwind direction (indicated by dash lines) are used to create nodal gradient vector.

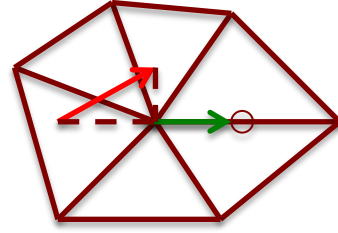


Figure 9. Weighted nodal gradient vector is used to extrapolate to mid-edge.

In anticipation of dynamic movement of boundaries a novel mesh-smoothing scheme has been developed based on mesh optimization techniques. The basic optimization method used condition number to define a cost function, C , for each element. [Freitag, Knupp] If the element is inverted the cost is based on the Jacobian, J , which will be negative. Otherwise the condition number is used. The condition number, CN , for an element is constructed from the product of the Frobenius norm of the matrix product of A and W and their inverses. A is the matrix whose columns are the edge vectors from the tetrahedron, shown in Figure 10. The weight matrix, W , is created by assembling the coordinates of the corners of an ideal element. This transforms a right-angled tetrahedron to a regular (equilateral) tetrahedron.

$$C = 1 - J, \text{ if } J \leq 0.0$$

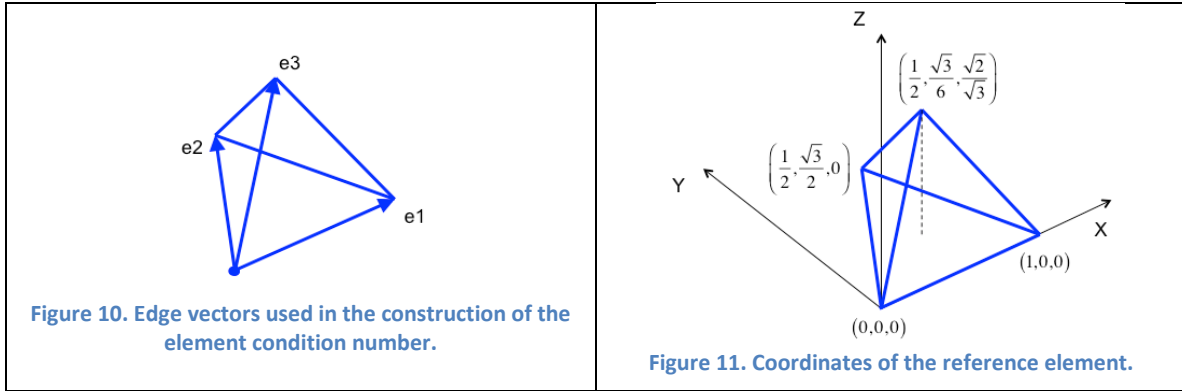
$$C = 1 - \frac{1}{CN}, \text{ if } J > 0.0$$

$$CN = \frac{\|AW^{-1}\| \|WA^{-1}\|}{3}$$

$$A = \begin{bmatrix} e1_x & e2_x & e3_x \\ e1_y & e2_y & e3_y \\ e1_z & e2_z & e3_z \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 1/2 & 1/2 \\ 0 & \sqrt{3}/2 & \sqrt{3}/6 \\ 0 & 0 & \sqrt{2}/\sqrt{3} \end{bmatrix}$$

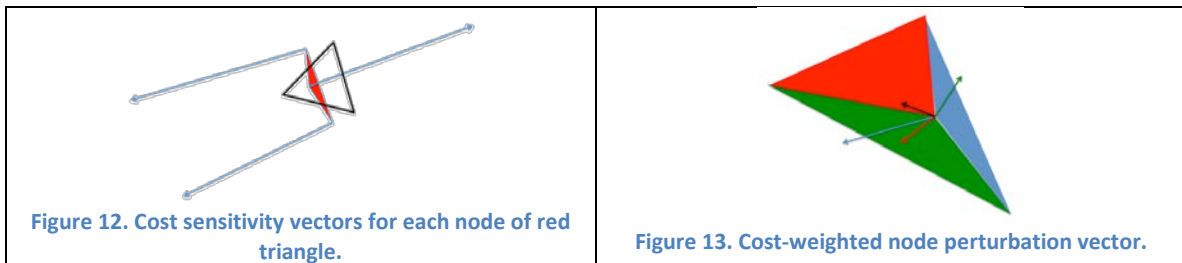
The condition, as defined, is unique to a tetrahedron irrespective of which corner is used. The cost function will be zero for an ideally shaped element, approach one as the element collapses and be greater than one for inverted elements. The goal of the mesh-smoothing scheme is to minimize the cost of each element.



Node-based cost values can be constructed by averaging the surrounding element cost values. Perturbing nodes in multiple directions and testing the new value of the node-based cost achieved optimization. This is extremely expensive.

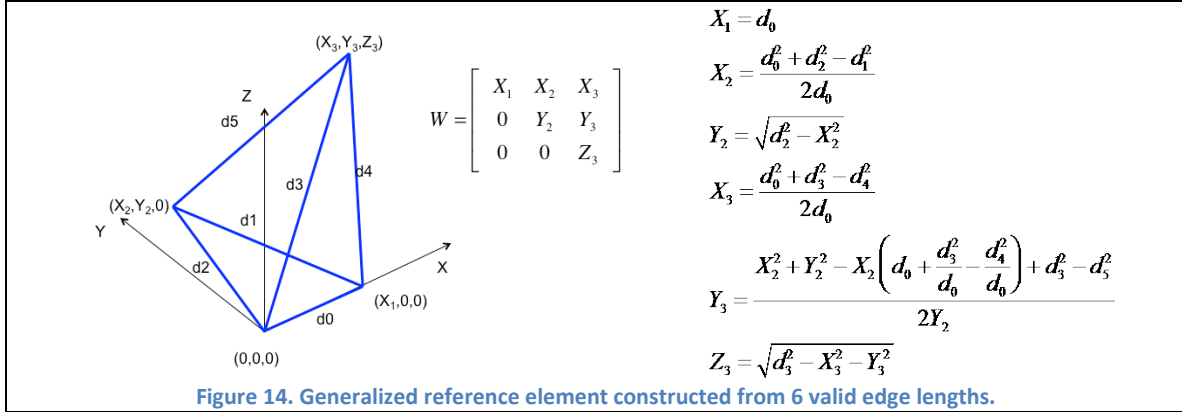
The modified approach developed under this grant uses sensitivity derivatives of element cost values with respect to each node in the element. This will be explained in two dimensions with an obvious extension to 3D. Figure 12 shows a high aspect ratio triangle in red. The vectors emanating from the corners represent the sensitivity vector of the element cost with respect to each node. The ideal triangle is shown in black. Moving the nodes in the directions of the arrows will increase the element cost. A nodal perturbation vector is computed using cost-weighted averages of the negative of the sensitivity vectors, shown in Figure 13. The perturbation from each triangle is color coded with the triangle color. The cost-weighted vector is shown in black. The formula for computing the perturbation vector at the node is shown below.

$$\vec{p}_n = - \frac{\sum_{e=1}^{ne} (\vec{s}_n)_e C_e}{\sum_{e=1}^{ne} C_e}$$

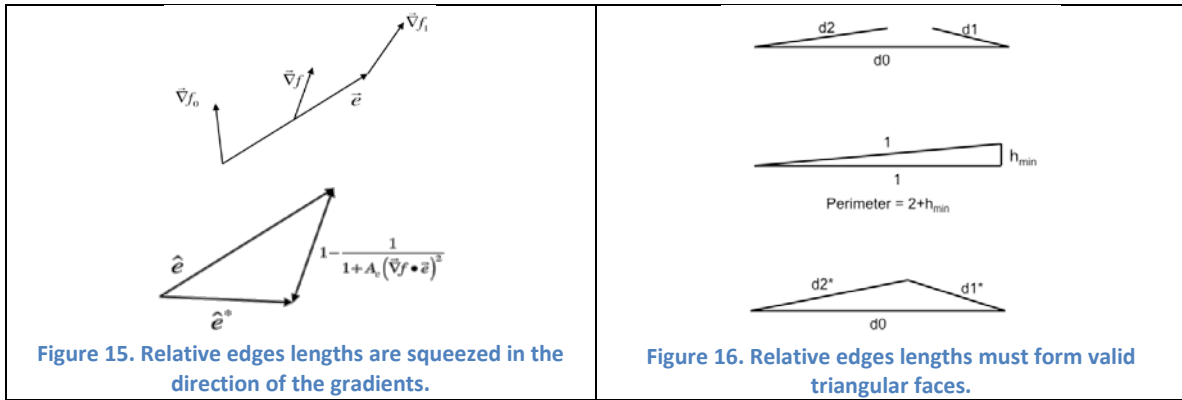


A perturbation vector is computed for each node. Nodes are perturbed incrementally in an iterative fashion. New perturbation vectors are computed each iteration. Smoothing using this new scheme is significantly faster than the previous perturb-test, perturb-test approach.

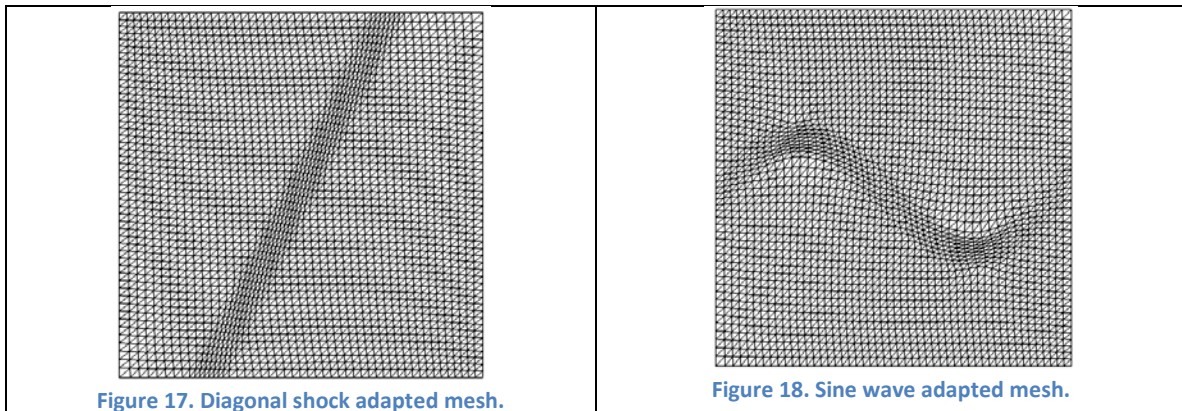
Adaptation was then added to the smoothing scheme by modifying the “ideal” element weight matrix. To begin, the coordinates of the reference element are generalized based on 6 valid edge lengths, d0-d5, shown in Figure 14. These edge lengths are relative, not absolute, with a maximum magnitude of 1. If all edges are unit length then the original W matrix is produced.



Next a scheme was devised to compute the relative edge lengths for all edges in the mesh. Gradients of the adaptation function are computed at the nodes and averaged to the mid-edge locations, shown in the upper portion of Figure 15. The normalized edge vector, \hat{e} , are squeezed in the direction of the edge gradient. If the gradient magnitude is zero or perpendicular to the edge no reduction in length occurs. A relative edge length for each edge in the mesh is computed in this manner. Next these relative edge lengths must be checked to ensure valid reference elements are constructed. This involves ensuring that any three relative edges of a triangle in the mesh must form a valid triangle, shown in Figure 16. This is achieved by ensuring the sum of the edge lengths from the triangle are greater than or equal to a minimum perimeter distance. The user specifies the h_{\min} value, typically on the order of 0.001. When this minimum perimeter distance is violated the shorter edges are extended to valid lengths. This is an iterative process over the entire mesh.



Examples of this type of adaptation are shown in Figure 17 and Figure 18. The domain is three-dimensional with one layer of elements in the Z direction (in/out of the page). The mesh on the Z maximum face is shown in each case. These are analytically defined gradient fields. The user controls the level of adaptation through the A_c coefficient in the edge squeezing formula.



An example of adapting to Mach number is shown in Figure 19. The mesh appears two-dimensional, but is three-dimensional with one cell layer in the Z direction. The top portion of the figure shows the Mach contours. The bottom portion shows the mesh on the Z maximum face. It is clearly evident that adaptation is taking place to the reflecting shocks and expansion fans in the flowfield.

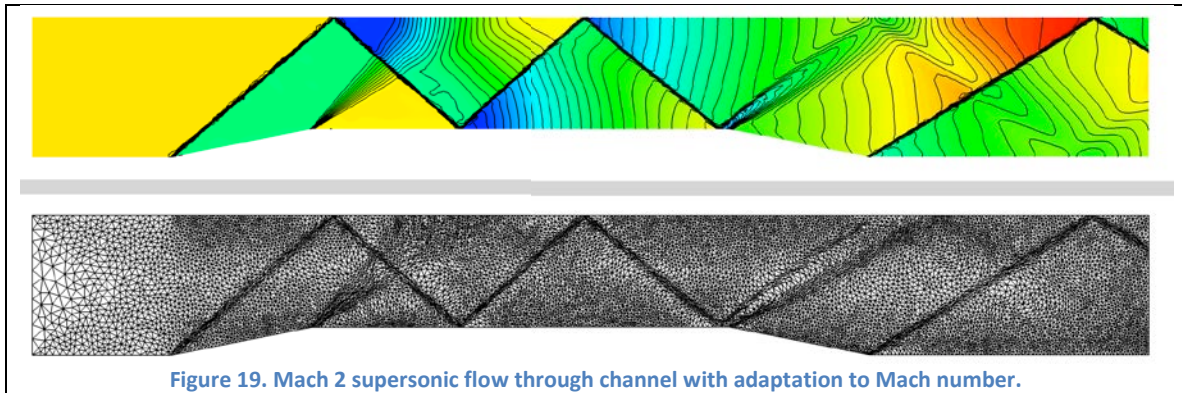


Figure 19. Mach 2 supersonic flow through channel with adaptation to Mach number.

This mode of mesh adaptation maintains a fixed number of nodes and fixed element connectivity.

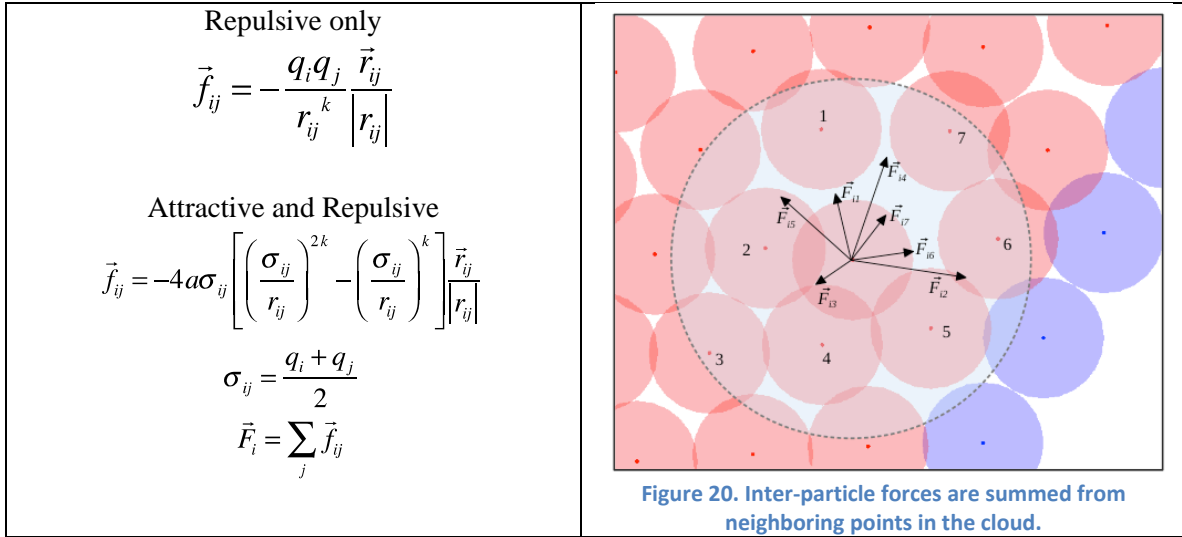
The baseline serial capability for TetFlow is operational. Validation cases are underway. The plans for the next phase include implementing an edge/face-flipping scheme via a tetrahedral mesh generation program to improve element quality as the bodies move through the domain. This has been tested on interior edges with great success. The ability to flip boundary face/edges is nearly complete. This tetrahedral meshing program can also be used to add and delete points. In addition the moving body capability will be implemented.

PointFlow

The code PointFlow has been developed as the test bed for untraditional meshless solvers. It is obviously node-centered and solves the differential form of the Euler equations using numerical differential quadrature. PointFlow generates the points and evolves the points as the solution develops. The code is currently first order in time and up to second order in space.

Geometry is supplied to PointFlow as a triangulated surface mesh, usually the same surface mesh used by OctFlow and TetFlow. The initial point set is created from the corner points of the surface mesh (called critical points), selected points along boundary edge curves, selected triangle face centers and centroid points from an Octree-based volume mesh. The initial processed evolves the mesh several steps before any flow solution commences.

Points are moved about via inter-particle forces between nodes. Two different force models are programmed. One involves repulsive forces based on Coulomb's law and the other is an attractive/repulsive force model following the Lennard-Jones pair potential. These two formulae are included next to Figure 20. The "desired" distance between points is defined by the q variable. The desired distance between two nodes i and j is defined using σ_{ij} . Each point has a collection of neighboring points in its cloud, shown as a dashed-line circle in the image. The total force on a node is the summation of the forces from the neighboring points in its cloud.



Points are iteratively moved in the direction of the summed force vector an incremental amount. The movement is governed by the equations of motion, shown below. We seek the steady-state position of the nodes so the drag coefficient and the starting velocity each step is set to zero. The mass is set to 1. The time-step is based on the proximity to neighboring points in a local time-stepping approach or set to some global minimum value.

$$F_i - C_D v_i = ma$$

$$F_i = m \frac{d^2 x_i}{dt^2}$$

$$d^2 x_i = F_i dt^2$$

$$\Delta x_i = \Delta t_i v_{i,0} + \frac{1}{2} \Delta t^2 F_i$$

$$\Delta x_i = \frac{\Delta t^2}{2} F_i$$

As points move about some will try to leave the domain. Interior points that pass through a boundary are repositioned on the closest boundary. Points on a given boundary that try to pass to a neighboring boundary are reposition on the curve between the two boundaries. And points located at the critical points of the geometry remained fixed to that position.

Point clouds are gathered for each node. The radius of the cloud is based on the closest neighbor distance multiplied by a user-defined parameter, typically in the range from 2 to 4. Interior points do not connect across the boundary to nodes on the other side. Boundary points do not connect to interior nodes they cannot “see”. Neighbor reciprocity is enforced where each point is in the cloud of its cloud points.

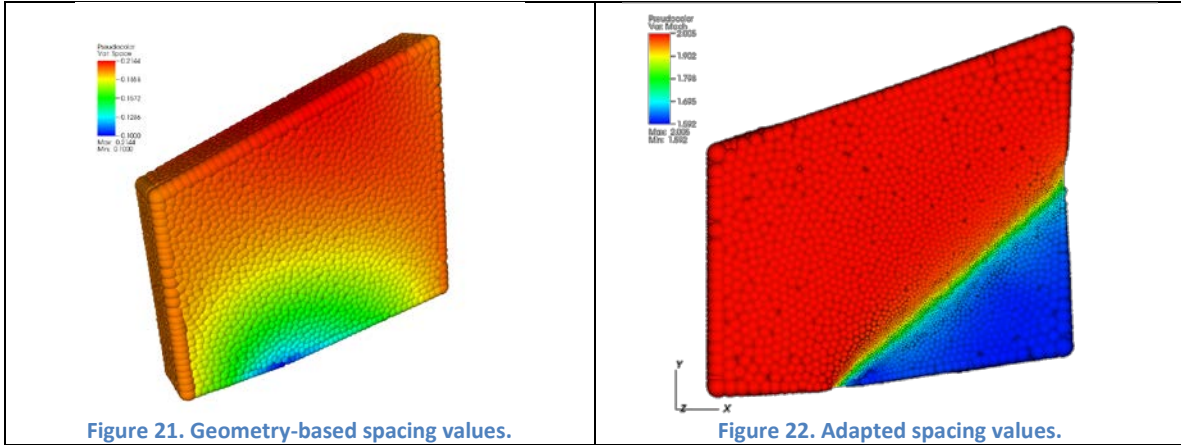
Spacing values travel with the nodes as they move. A geometry-based spacing value, S_g , is constructed based on a blending of critical points spacing values, which are controlled by the user. The blending is simply an inverse-distance weighted average of the neighbors in the current point cloud. An adaptation delta, ΔA_n , is combined with the geometry spacing value to define the final nodal spacing, S_n , shown below. Adaptation is based on the magnitude of the gradient of the user-selected function, such as density, pressure, velocity magnitude or Mach number. A threshold value of the gradient magnitude is computed from the mean and standard deviation values of the nodal gradient magnitudes. The adaptation delta is then the minimum of 1 and the ratio of the node gradient magnitude and the threshold. This delta multiplies the difference between the geometry spacing and the defined minimum spacing and is subtracted from the geometry spacing.

$$|\nabla f|_{Threshold} = |\nabla f|_{Mean} + C\sigma$$

$$\Delta A_n = \text{Min}\left(1, \frac{|\nabla f|_n}{|\nabla f|_{Threshold}}\right)$$

$$S_n = S_g - \Delta A_n(S_g - S_{min})$$

The geometry-based spacing values for a ramp test case are shown in Figure 21. The size and color of these spheres is the geometry spacing value. As the solution develops the gradient of the selected adaptation functions evolve. The final adapted spacing values for the same case are shown in Figure 22. The size and color of these sphere is the final adapted spacing value.



Points are added and deleted based on an overlap ratio. Overlap ratio measure the degree to which the spacing between each node is met. They are computed for each node and involve on the neighbors that are at a distance of 1.1 times desired distance between points, σ_{ij} . Edge nodes are tested against other nodes on the same edge. Surface nodes are tested against other nodes on the same surface. Interior nodes are tested against all of its neighbors with the specified distance. The

overlap ratio formulae are given below. They differ depending on the node characteristic; i.e. edge, surface or interior node. These are approximate formulae.

$$OVR_e = \sum \frac{\sigma_{ij}}{2\Delta s_{ij}}, \text{ normalized distance between nodes}$$

$$OVR_s = \sum \left[\frac{q_j}{\pi q_i / 4} \frac{r}{\Delta s_{ij}} \right], \quad r = \frac{q_i}{2}, \text{ normalized projected length}$$

$$OVR = \sum \left[\frac{\pi q_j^2 / 4}{\pi q_i^2 / 4} \frac{r}{\Delta s_{ij}} \right], \quad r = \frac{q_i}{2}, \text{ normalized projected surface}$$

$$\sigma_{ij} = \frac{q_i + q_j}{2}$$

Figure 23 is used to define the overlap ratios for edges and surface nodes. Edge node overlap ratios are basically normalized distances between nodes. When the nodes are at the proper distance the ratio will be near unity. Surface node overlap ratios are normalized projections of the spacing value projected onto the circle for the central node. The image on the right shows the projections of the neighbor bubble onto the central node, green circle. The formulae take into account different desired spacing values between the nodes. Again a unit value of the overlap ratio indicates the nodes are at the proper distance and have good coverage of the space. The extension of the method to interior nodes results in the normalized projection of the bubbles onto the sphere surrounding the central node.

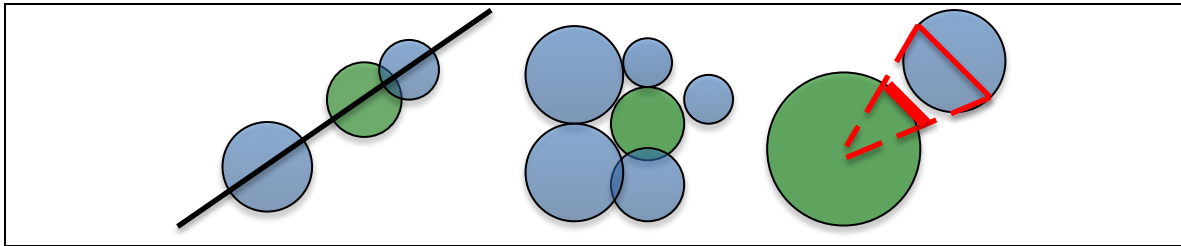


Figure 23. Overlap ratios for line and surface.

During the refinement and deletion process nodes with an overlap ratio greater than a specified value, such as 3 or 4, are deleted. Nodes with an overlap ratio less than a specified value, such as 0.25 or 0.5, are tagged and new neighbors are created near those points. Nodes with overlap ratios between these values are unchanged.

The inviscid flow solution is computed using differential quadrature. The differential form of the Euler equations, shown below, is solved numerically using a first order backward differenced time integration scheme. The weight coefficients, w_x , w_y and w_z , can be computed using either Basis Functions or Least Squares. The weights are vectors between the central node and its neighbors. The scheme can be conservative if certain properties are met. [Chiu, Wang, Jameson]

Construction of the weights for a conservative scheme involves establishing a global system of equations and solving for the conservative weights. The solution of the global system can be expensive and would be impractical for the moving body cases if it was required each step. The resulting weights have the appearance of area vectors between nodes divided by a volume like term, analogous to a Finite-Volume discretization. An alternative was explored in two dimensions where a triangular mesh was constructed, the appropriate real Finite-Volume weights created and then the mesh was discarded. A similar approach will be implemented 3D once the tetrahedral mesher is fully operational.

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} F(Q) + \frac{\partial}{\partial y} G(Q) + \frac{\partial}{\partial z} H(Q) = 0$$

$$\frac{\Delta Q}{\Delta t} = - \left[\sum_j^{nc} w_x^j \Delta F_j + \sum_j^{nc} w_y^j \Delta G_j + \sum_j^{nc} w_z^j \Delta H_j \right]$$

Some comments on the use of a tetrahedral mesh program in a “meshless” method are needed. There is truly no such thing as a meshless method. The point clouds provide connectivity to neighboring nodes, which is a form of a mesh. The tetrahedral mesh generator produces elements, which thereby provide connectivity to neighboring nodes as well. But the tetrahedral mesh version provides a much more efficient set of neighbor connections. It will produce significantly fewer neighboring points and will provide proper coverage of the space surrounding each point. And the direction of the research will be such that small regions of the space can be tessellated with the tetrahedral mesh program and processed for the weights instead of the entire domain. This is especially true for parallel.

The weights operate on fluxes computed at the midpoint between the central node and its neighbors. This is shown in 2D in Figure 24 between two nodes. There associated cloud of neighbors is also shown. Second order spacial accuracy is achieved by extrapolating primitive variables to this midpoint location from upwind directions, i.e. left and right directions depicted in Figure 25. The gradient vectors $\vec{\nabla} q_i$ and $\vec{\nabla} q_j$ are computed using weighted averages of gradients from neighbors in the upwind directions, shown in blue and red in the figure. A MinMod type flux limiter is used to prevent extrapolations to negative pressure or density.

$$q_L = q_i + \vec{\nabla} q_i \frac{\vec{r}_{ij}}{2}$$

$$q_R = q_j - \vec{\nabla} q_j \frac{\vec{r}_{ij}}{2}$$

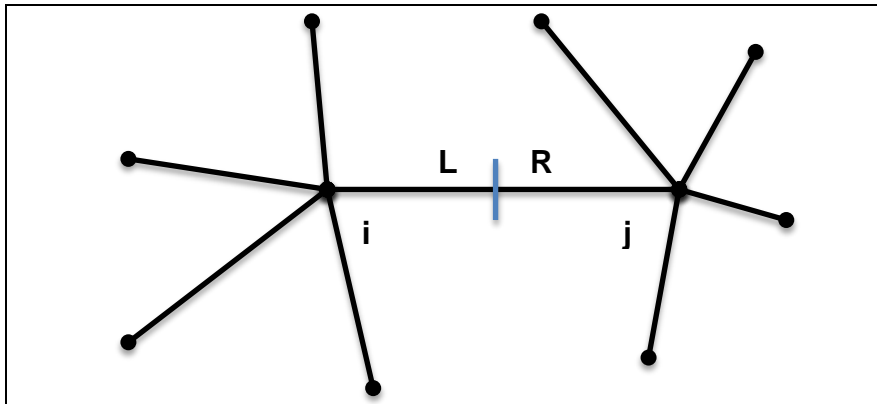


Figure 24. Mid-point location between nodes where flux is constructed.

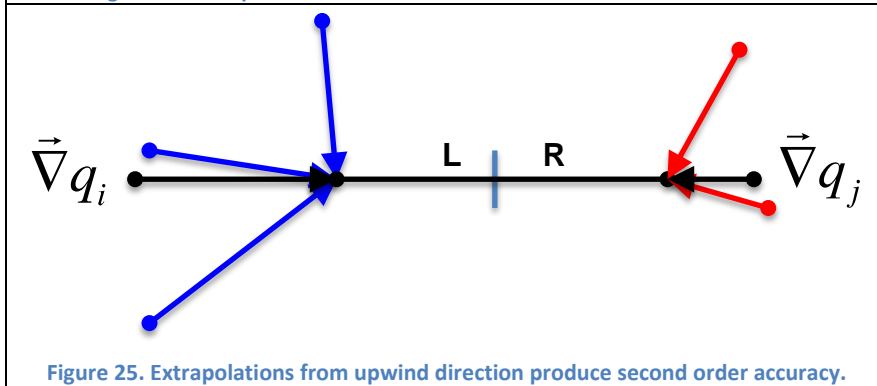


Figure 25. Extrapolations from upwind direction produce second order accuracy.

The baseline serial capability for PointFlow is operational. Validation cases are underway. The plans for the next phase include implementing Finite-Volume version of the weights by using the tetrahedral mesh program. A second order implicit BDF2 scheme will be installed. In addition the moving point capability will be implemented.

Validation Cases

Several cases are being used to validate the flow solvers and mesh adaptation schemes. These are relatively simple configurations and are being solved on small workstations. The preliminary results for each case and each code are shown below.

Supersonic ramp:

The first case is a Mach 2 flow over a 10-degree wedge, shown in Figure 26. The geometry appears two dimensionally, but is actually three dimensional by extension in the Z direction (out of the paper). This case has one distinct feature, the shock, which all three codes capture. Adaptation is performed based on the gradients of Mach number. OctFlow and TetFlow show very crisp resolution of the shock. OctFlow refines heavily in the shock region. TetFlow squeezes the mesh in the normal direction of the shock. PointFlow redistributes points near the shock and reduces the spacing value at the shock, but shows some smearing of the shock, most likely due to

the Least-Squares quadrature and large point clouds. PointFlow was unstable if a smaller point cloud radius was used, as this caused some point clouds to be biased to one side with improper coverage of the space.

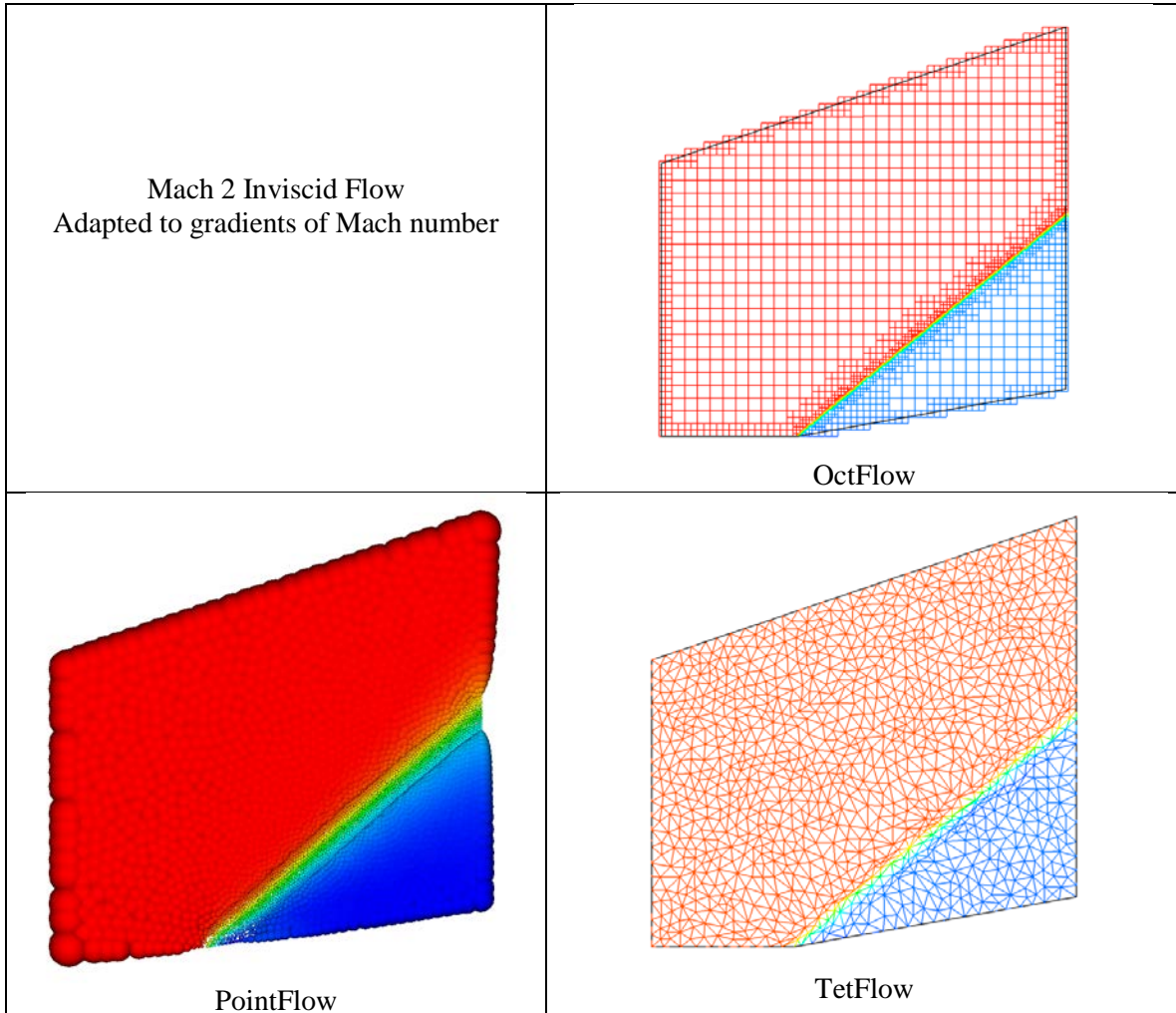


Figure 26. Supersonic flow over 10-degree wedge.

Onera M6:

The second case is the Onera M6 wing in transonic flow. This is a popular validation case and provides an opportunity to compare results from the codes with experimental data. Figure 27 shows results from the three codes. The color and line contours are for pressure. The OctFlow and TetFlow results are plotted on the same scales. PointFlow has preliminary results that are 1st order in space.

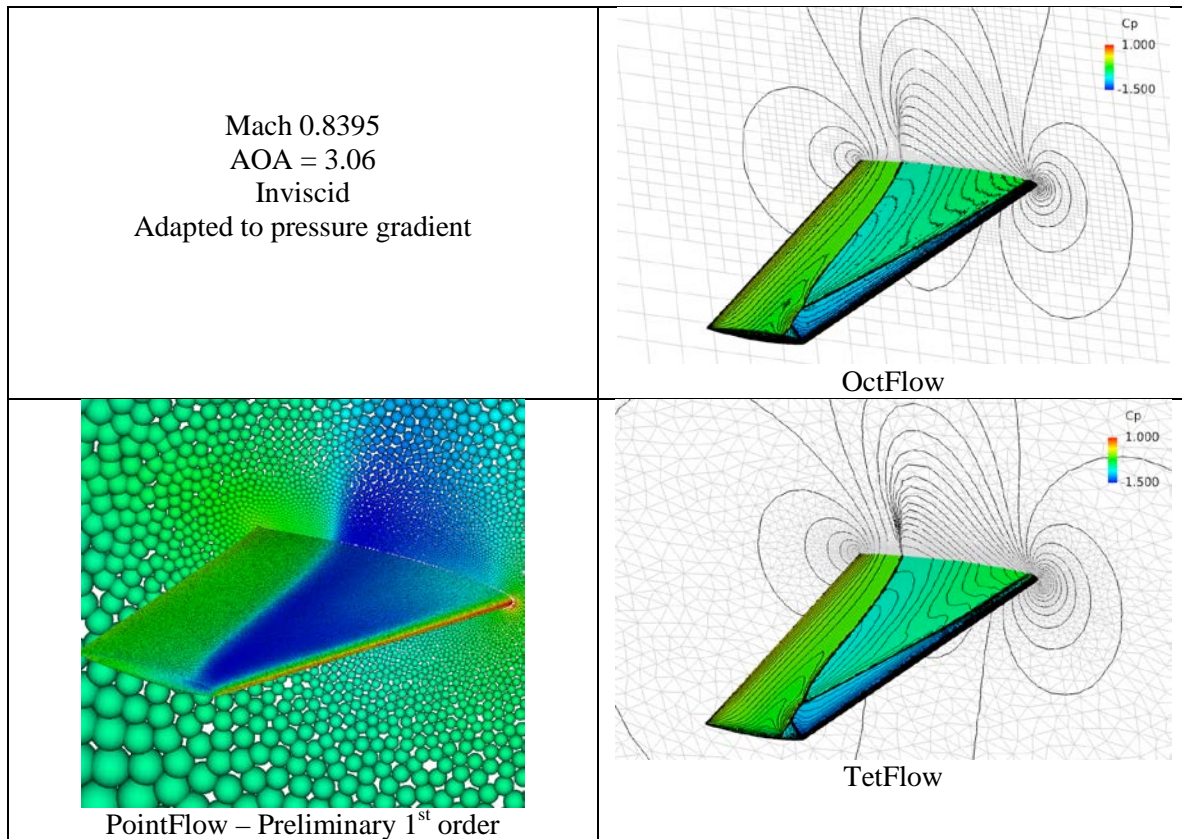


Figure 27. Onera M6 transonic case

The mesh adaptation from TetFlow is displayed in Figure 28 and Figure 29. Adaptation of the elements near the shock is evident. Alignment of edges on the symmetry plane ahead of the shock is also evident.

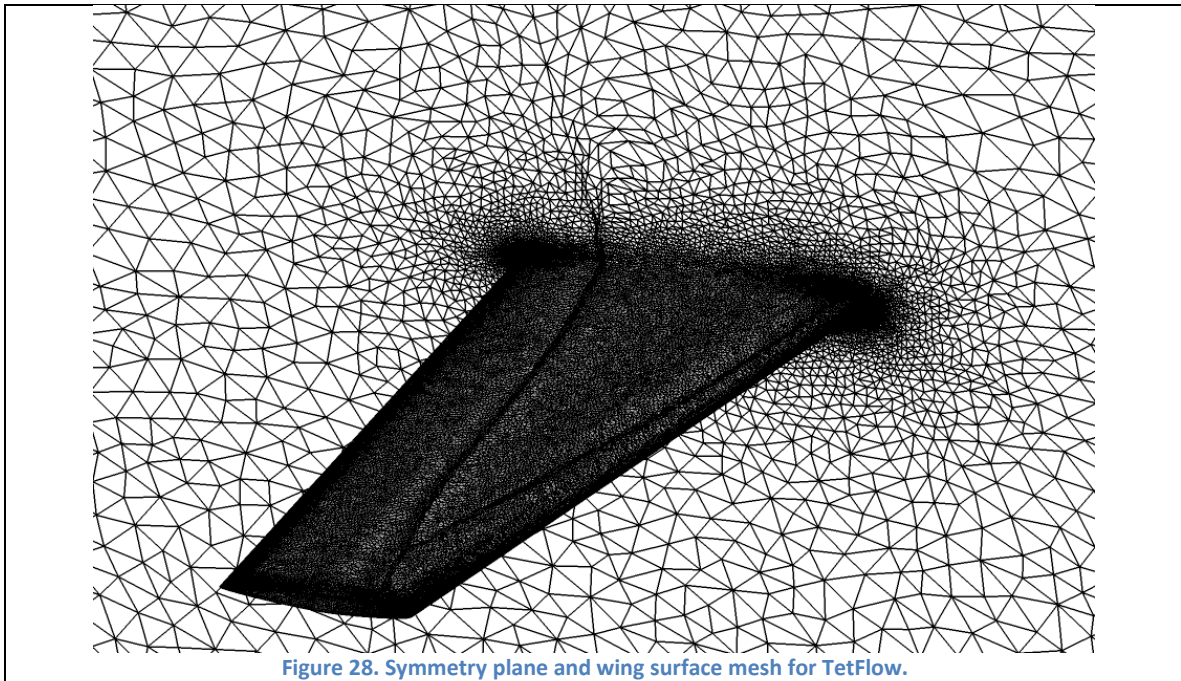


Figure 28. Symmetry plane and wing surface mesh for TetFlow.

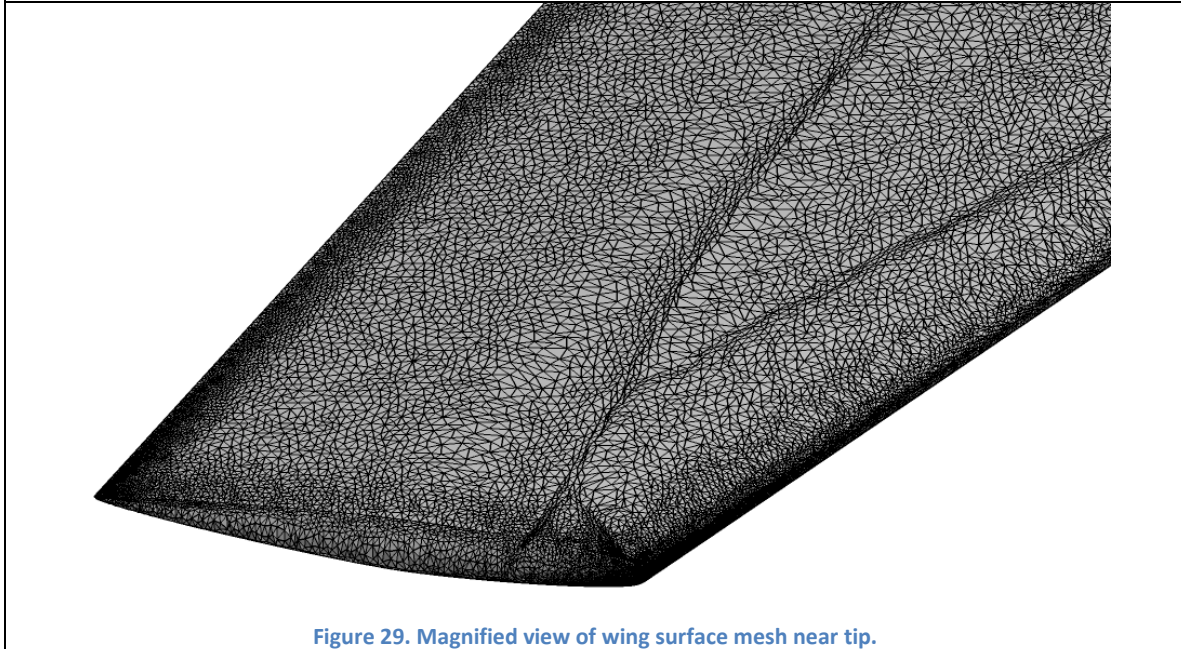


Figure 29. Magnified view of wing surface mesh near tip.

A preliminary look at the spacing parameters for PointFlow is shown in Figure 30. The spheres are sized using the geometry spacing. The colors for the images on the left are modified to highlight the differences in the spacing values for the nodes near and on the surface. The adaptive spacing delta values clearly identify where the shock is located.

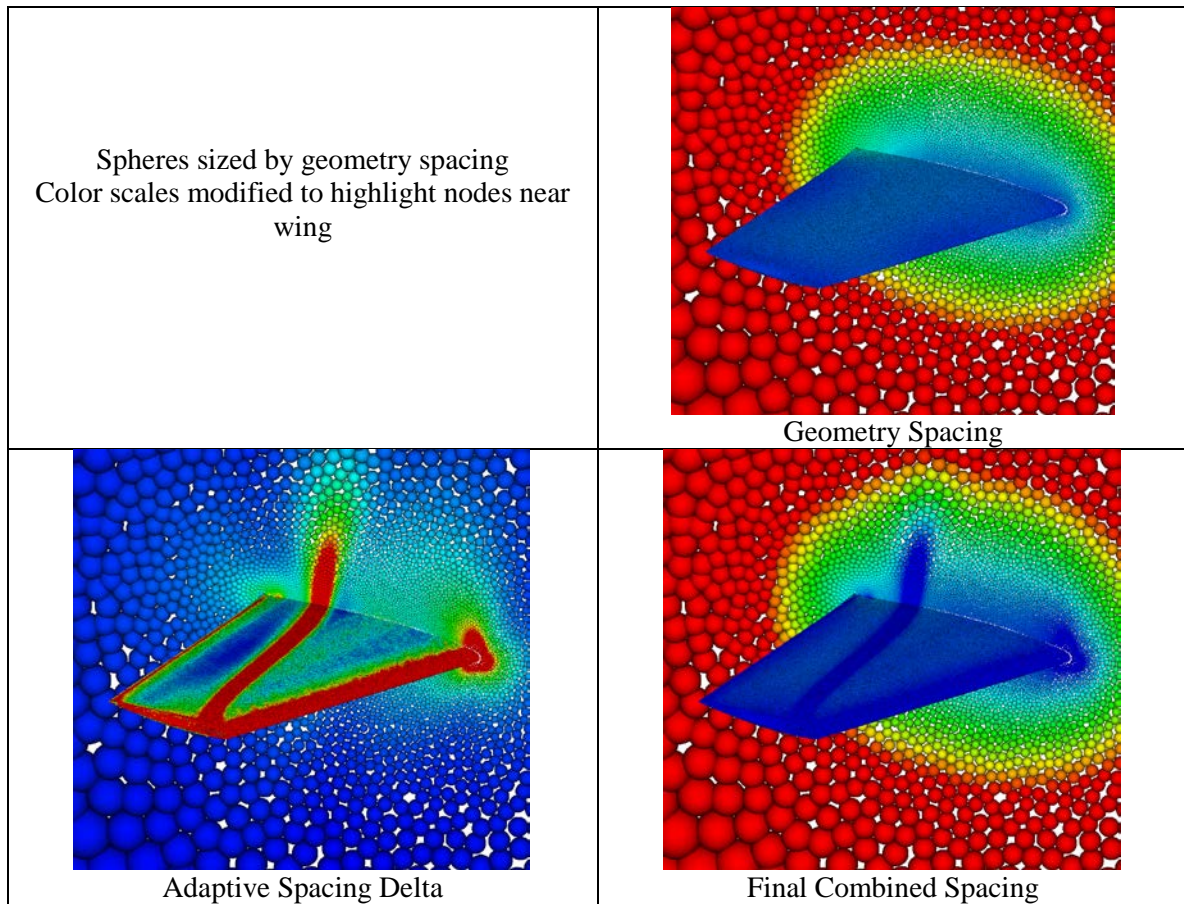
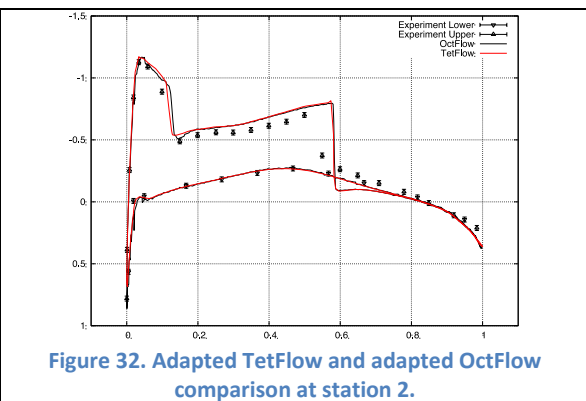
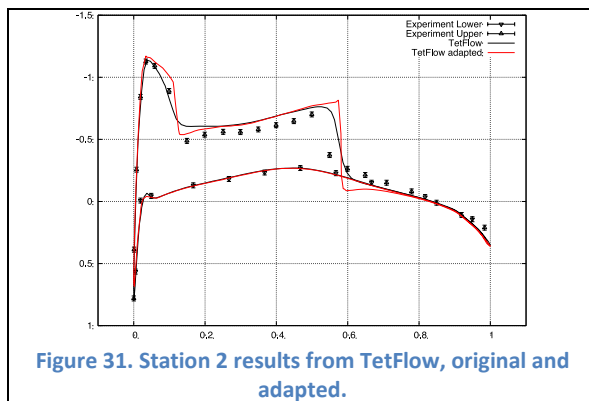


Figure 30. Preliminary results of PointFlow size parameters for M6.

Comparisons of TetFlow and OctFlow results with experimental data are shown in Figure 31 and Figure 32. The shock is crisply resolved in both cases and the codes compare well with each other. Comparison with data is considered very good for inviscid solutions.



Eglin Wing/Pylon/Store:

The final case is the Eglin wing with a pylon and store. This is another popular validation case. It has experimental force and pressure data for a prescribed motion of the store on the sting mount. Results from the three codes are shown in Figure 33. Color contours of pressure are shown. The contour scales for OctFlow and TetFlow are the same. The PointFlow results are preliminary and 1st order in space.

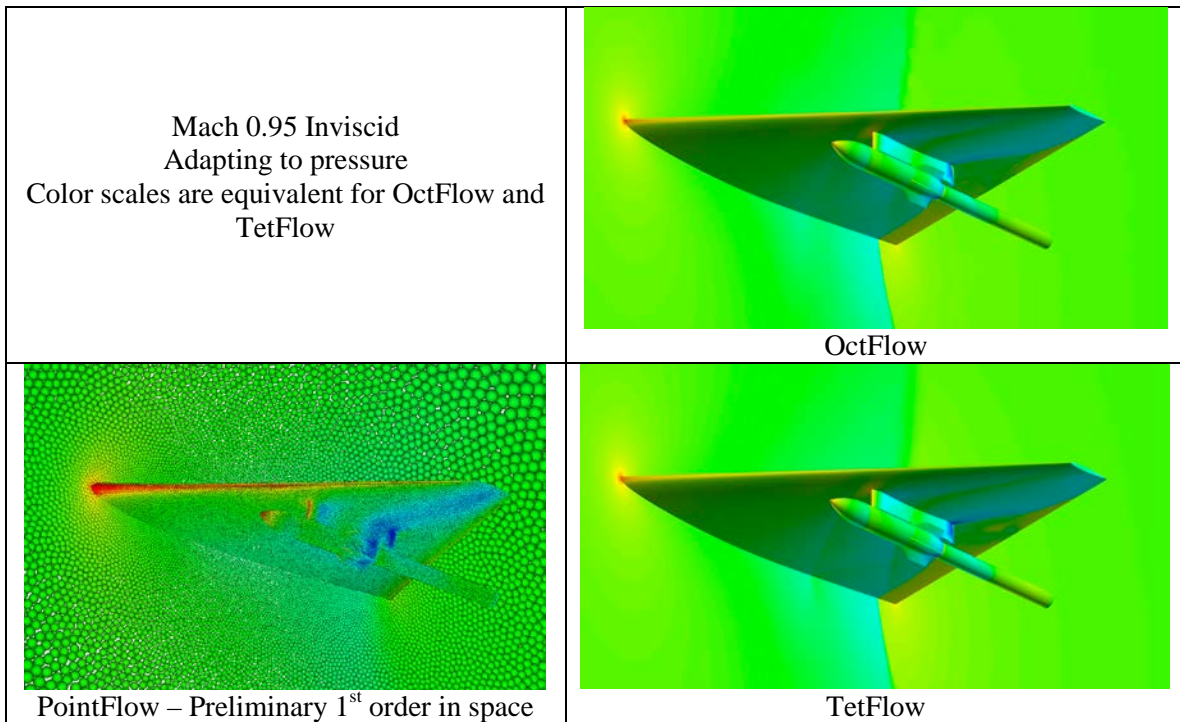


Figure 33. Eglin Wing/Pylon/Store comparisons.

Several views of the mesh adaptation within TetFlow are shown in Figure 34. Shock structures are evident without displaying the solution contours. There is a reflected shock shown on the underside of the wing near the symmetry plane. This illustrates the adaptation that is taking place throughout the mesh, not just at the strong shock regions.

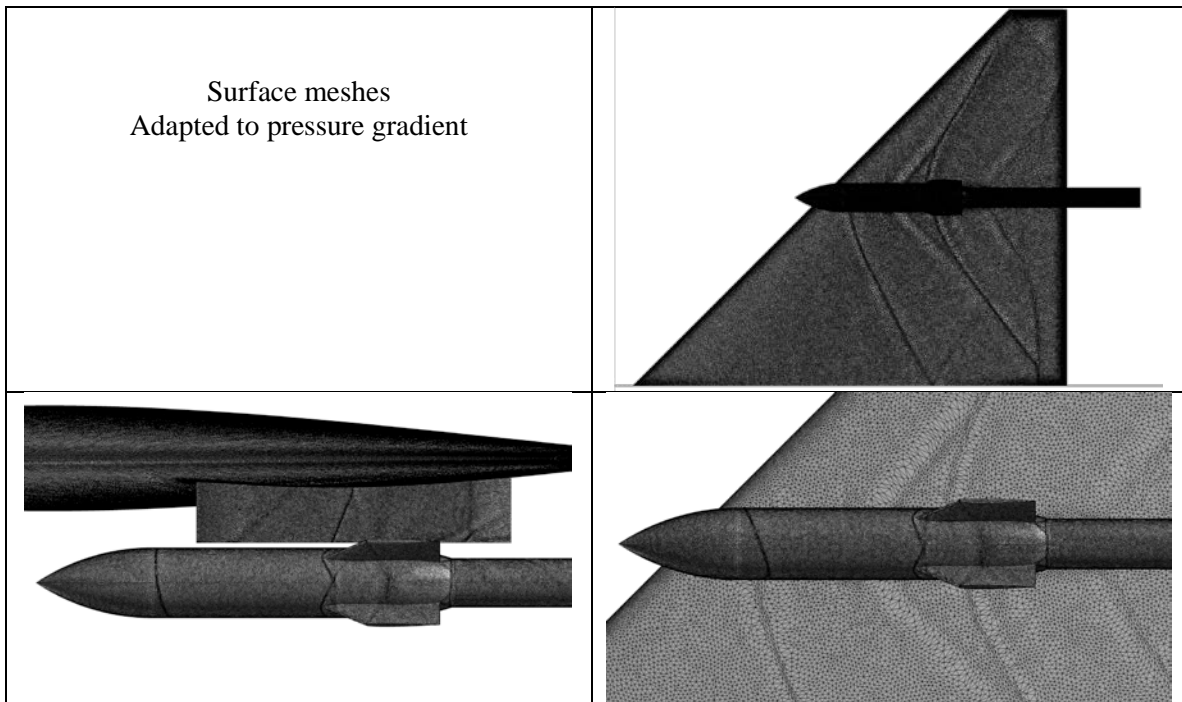
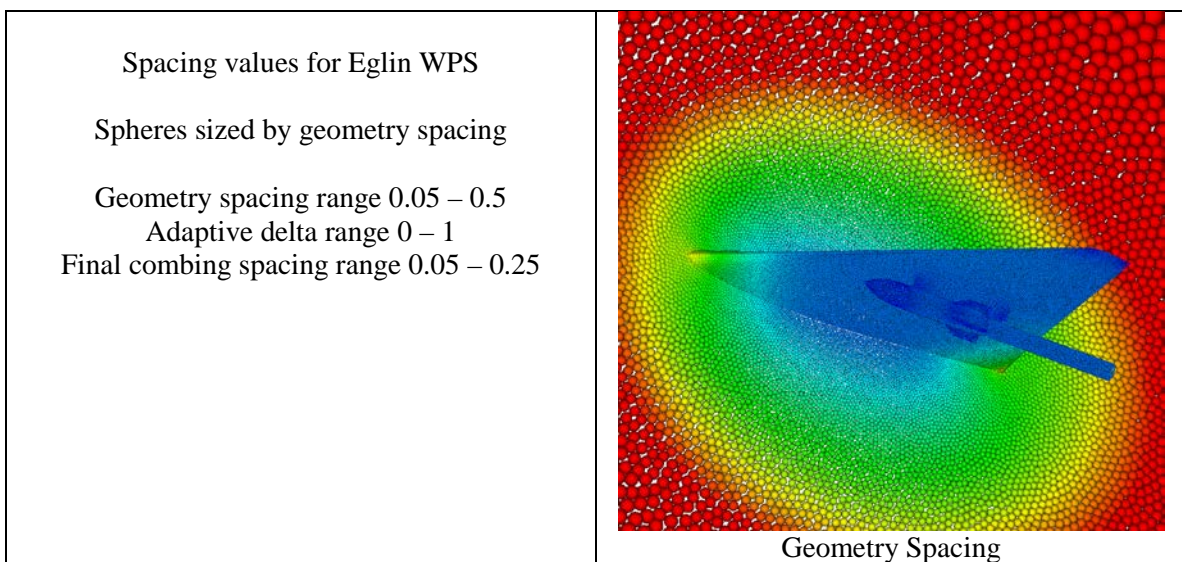


Figure 34. Various views of the surface meshes adapted to pressure gradient.

A preliminary look at the spacing parameters for PointFlow is shown in Figure 35. The spheres are sized by the geometry spacing values. The color ranges are different for each plot to highlight the nodes near the wing and store. The adaptive delta values are identifying the high pressure gradient regions near the leading and trailing edges and, to a lesser extent, the shock regions. The PointFlow solution is preliminary and 1st order accurate in space.



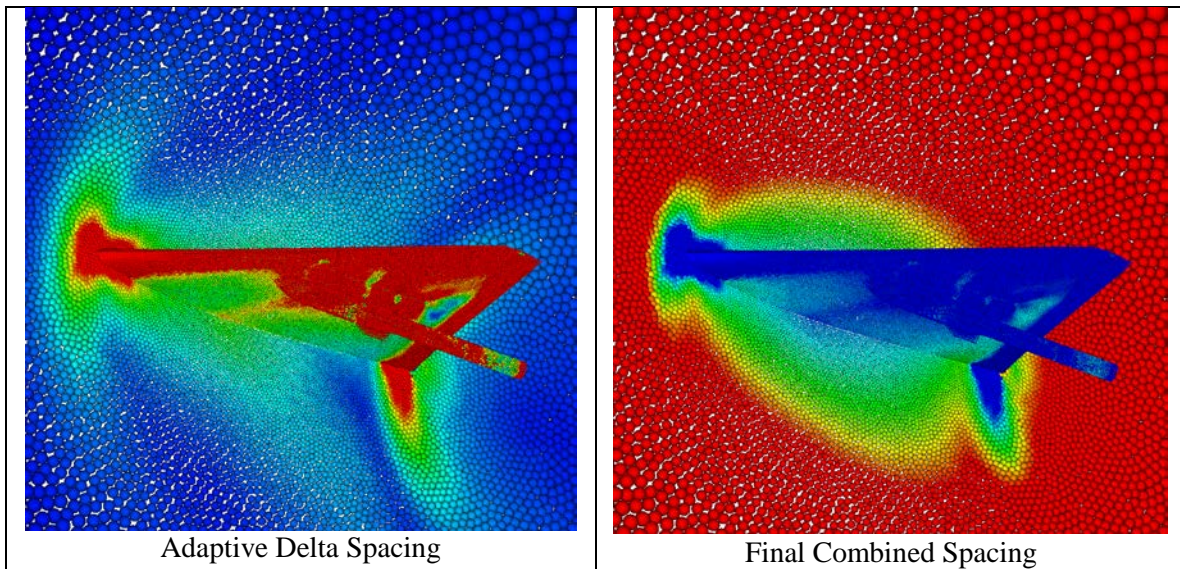


Figure 35. Various spacing values for Eglin WPS PointFlow solution.

3. Significance of Results

The goals for the first year are to have working serial versions of all three codes for simulating pseudo-steady analyses. All three codes are at that state of capability, although only TetFlow has enhanced time integration through second order implicit time differencing.

Validation cases are under way using the three codes to document the performance of the three methods and the benefits of the mesh adaptation techniques. The current cases have known analytical or experimental data to compare with. A combined carrier/aircraft case is under development to allow the pseudo-steady analysis of the landing trajectory using all three codes.

4. Plans and upcoming events for next reporting period

The second year of the project will focus on the time accurate simulation of body movement through the domain. This will require completing the implementation of grid speed terms and Geometric Conservation Law enforcement in all codes.

The adaptive mesh smoothing technology will be presented at the AIAA conference in January in Kissimmee Florida. This will also be submitted to the AIAA Journal for publication.

5. Recommended reading

References mentioned in this report are listed below.

Shipman, J.D., Arunajatesan, P.A., Cavallo, P.A., Sinha, N. and Polsky, S.A., "Dynamic CFD Simulation of Aircraft Recovery to an Aircraft Carrier", AIAA-2008-6227.

Biedron, Robert T., and Thomas, James L., “Recent Enhancements To The FUN3D Flow Solver For Moving-Mesh Applications”, AIAA-2009-1360.

Biedron, Robert T., Vatsa, Veer N., and Atkins, Harold L., “Simulation of Unsteady Flows Using an Unstructured Navier-Stokes Solver on Moving and Stationary Grids”, AIAA-2005-5093.

Freitag, Lori A., and Knupp, Patrick M., “Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number”, 8th International Meshing Roundtable, Paper 5a3, 1999.

Chiu, Edmond Kwan-yu, Wang, Qiqi, and Jameson, Antony, “A Conservative Meshless Scheme: General Order Formulation and Application to Euler Equations”, AIAA-2011-0651.

6. Transitions/Impact

The new mesh optimization scheme developed for TetFlow has been extended to hybrid meshes and incorporated into the production mesh-smoothing tool, P_OPT, used at the SimCenter. Master’s student, Yanan Gong, in her research on time dependent moving body simulations, also uses this improved method and will defend her thesis in November.

The point generation scheme used by PointFlow will be implemented by Ph.D. student Philip Fackler to provide points to the tetrahedral mesh code for general use at the SimCenter.

7. Collaborations

Technology used by OctFlow is frequently discussed with Matthew O’Connell, a UTC Ph.D. student employed at NASA Langley as a coop student. He is researching approaches to perform large-scale parallel simulations.

8. Personnel supported

Principal investigator: Dr. Steve Karman

Co-investigator or Co-PI: N/A

Post Doctoral students: N/A

Graduate Students: Yanan Gong, Philip Fackler, Tuo Liu

Undergraduate Students: N/A

Research staff: Bruce Hilbert

Other: N/A

9. Publications

Publications resulting from this project:

Conference Papers

Abstract accepted to American Institute of Aeronautics and Astronautics, “Adaptive Optimization-Based Smoothing for Tetrahedral Meshes”, submitted to SciTech 2015, Kissimmee FL.

10. Point of Contact in Navy

Judah Milgram, ONR, October 8, 2014.

John Kinzer, ONR, September 24, 2014.

11. Acknowledgement/Disclaimer

This work was sponsored by the Office of Naval Research, ONR, under grant/contract number N00014-14-1-0010. Authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Office of Naval Research, or the U.S. government.

Section II: Project Metrics

Grant # N00014-14-1-0010

Adaptive Meshing of Ship Air-Wake Flowfields

Progress Report for Period: October 1, 2013 – September 30, 2014

PI: Dr. Steve Karman

(423) 425-5492

Steve-Karman@utc.edu

University of Tennessee at Chattanooga

Date Prepared: October 21, 2014

Metrics

[Please include each of the following metrics. If none, please indicate N/A.]

Number of faculty supported under this project during this reporting period: 1

Number of post-doctoral researchers supported under this project during this period: N/A

Number of graduate students supported under this project during this reporting period: 2

Number of undergraduate students supported under this project during this period: N/A

Number of refereed publications during this reporting period for which at least 1/3 of the work was done under this effort:

Number of publications (all) during this reporting period

Number of patents during this reporting period: N/A

Number of M.S. students graduated during this reporting period: 1

Number of Ph.D. students graduated during this reporting period: 2

Awards received during this reporting period: N/A

Invited talks given: N/A

Conferences at which presentations were given (not including invited talks above): N/A

11. Financial information

[If this project received more than one source of funding, for example a DURIP grant for equipment and a research grant that funded personnel, please list the sources of funding]

[If elements of this project are funded by different agencies, this should be made clear at the project initiation, and should be reported in every progress report. Include a description of any other agency funding received for work reported in this progress report here.]

[If students working on this project are supported through a fellowship other than this grant, such as an NSF or NSDEG fellowship, or a military scholarship, please identify the source of student support here.]

[For this period: expenditures, subcontractor obligations and execution status, funding increments received. For obligations (funding received) and expenditures, use a table with format like this.]

FY 2014	Total Budget	Obligated This Period	Obligated Cumulative	Expended This Period	Expended Cumulative	Grant/ Contract Period of Performance
6.2 (Applied Research Funding)	98,474.00	98,474.00	98,474.00	80,621.00	80,621.00 \$10,392.17 is still outstanding through Sept 2014 and is not reflected in the above total.	10/01/2013-09/30/2014

[Use a separate row for each kind of funding (6.1, 6.2). Most projects will have only 6.1 or 6.2, and thus only one row. If only one type of funding is shown, a “Total” row is not required.]

12. Administrative notes and other items of interest

[Among other things, a good place to report interactions with other institutions or companies related to future projects and technology transition. Promotions received, new personnel, retirements, where students who have graduated are currently working, etc.]

